LIST OF a11y FIXES MADE

1. Skip Link Implementation

What the fix is

A skip link is an anchor placed at the top of a page that lets keyboard users jump directly to the main content. This prevents users from having to tab through the entire navigation every time a new page loads.

Functions

- Helps users with mobility impairments who cannot repeatedly press Tab.
- Helps screen-reader users quickly bypass repetitive navigation.
- Required by WCAG 2.1 (Success Criterion 2.4.1 – Bypass Blocks).

How it is used in Solar Consult

In Solar Consult, we implemented:

```
<a href="#maincontent">Skip to main content</a>
```

And:

```
<main id="maincontent">
  …
</main>
```

This ensures:

- The first Tab press focuses the skip link.
- Pressing Enter jumps directly to the main content.
- Focus correctly lands in the main section without JavaScript.

Impact on Solar Consult:
Improves navigation efficiency and makes the page immediately usable by keyboard-only users and screen readers.

2. Logical, Predictable Keyboard Tab Order

What the fix is

Ensuring that pressing Tab moves through interactive elements in a logical, intuitive sequence usually matching the visual layout and reading order.

Functions

- Keyboard-only users rely entirely on the Tab key.
- Disorganized tab order causes confusion and makes the site unusable.
- WCAG 2.1 (2.4.3 – Focus Order) requires a meaningful sequence.

How it is used in Solar Consult

The HTML is so that interactive elements appear in a clean top-to-bottom order, for example:

```
<nav>
  <a href="/about">About</a>
  <a href="/services">Services</a>
  <a href="/contact">Contact</a>
</nav>

<main id="maincontent">
  <form>…</form>
</main>
```

The browser automatically uses DOM order, so users tab through:

1. Skip link
2. Navigation links
3. Main content
4. Form fields
5. Buttons

Impact on Solar Consult:
Users, especially those with motor impairments or on screen readers can predictably navigate through your site without confusion or unexpected jumps.

3. Proper Form Labeling + Native HTML5 Validation

What the fix is

A HTML form needs the following:

- Correct <label> elements
- Proper input types (type="email")
- Required fields (required)
- Native browser error messages for invalid or missing information

Why it matters

Forms are one of the biggest accessibility barriers.
Users who cannot see the input field need label associations.
Users who cannot use a mouse need keyboard-friendly validation.

How it is used in Solar Consult

Our form contains:

```
<label for="fullname">Full name</label>
<input id="fullname" type="text" required>
```

```
<label for="email">Email</label>
<input id="email" type="email" required>
```

And the submit button:

```
<button type="submit">Submit</button>
```

This results in:

- Screen readers announcing “Full name, edit text” correctly.
- Keyboard users able to type and tab through fields.
- Browser showing native validation messages when email is empty or invalid.

Impact on Solar Consult:
Makes the form more usable for:

- screen-reader users
- keyboard-only users
- people with cognitive difficulties
- people with low vision

It also prevents incomplete submissions, which reduces user frustration.

4. Ensuring All Interactive Elements Are Actual HTML Elements (Links, Buttons, Inputs)

What the fix is

We have aoided using non-sematic html components such as:

- <div> or <span> as buttons
- Clickable areas with only JavaScript handlers
- Custom keyboard scripting

Instead, we used native interactive elements such as:

- <a> for navigation
- <button> for actions
- <input> for user input

Why it matters

Native elements come with built-in:

- Focus behavior
- Keyboard behavior
- Assistive technology support
- Accessibility roles

All without extra code.

How it is used in Solar Consult

In our project, it helps us achieve the following:

- Navigation uses real links:
- <a href="/about">About</a>
- Form uses real inputs and a real submit button:
- <button type="submit">Submit</button>
- No onclick scripts or pseudo-buttons created with <div>.

Impact on Solar Consult:
The site becomes automatically:

- Keyboard-friendly
- Screen-reader friendly
- Mobile-browser friendly
- Accessible without needing ARIA or JavaScript fixes

This reduces the risk of major A11y violations.